1.

| | |
|---|---|
| Record Nr. | UNINA9910963637503321 |
| Autore | Kowalik Janusz S |
| Titolo | Using OpenCL : programming massively parallel computers / / Janusz Kowalik and Tadeusz Puzniakowski |
| Pubbl/distr/stampa | Amsterdam, : IOS Press, c2012 |
| ISBN | 1-299-33347-8<br>1-61499-030-1 |
| Edizione | [1st ed.] |
| Descrizione fisica | 1 online resource (312 p.) |
| Collana | Advances in parallel computing ; ; v. 21 |
| Altri autori (Persone) | PuzniakowskiTadeusz |
| Disciplina | 005.2752 |
| Soggetti | OpenCL (Computer program language)<br>Parallel computers<br>Parallel programming (Computer science) |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | Description based upon print version of record. |
| Nota di bibliografia | Includes bibliographical references and index. |
| Nota di contenuto | Title Page; Preface; Contents; Introduction; Existing Standard Parallel Programming Systems; MPI; OpenMP; Two Parallelization Strategies: Data Parallelism and Task Parallelism; Data Parallelism; Task Parallelism; Example; History and Goals of OpenCL; Origins of Using GPU in General Purpose Computing; Short History of OpenCL; Heterogeneous Computer Memories and Data Transfer; Heterogeneous Computer Memories; Data Transfer; The Fourth Generation CUDA; Host Code; Phase a. Initialization and Creating Context; Phase b. Kernel Creation, Compilation and Preparations for Kernel Execution<br>Phase c. Creating Command Queues and Kernel ExecutionFinalization and Releasing Resource; Applications of Heterogeneous Computing; Accelerating Scientific/Engineering Applications; Conjugate Gradient Method; Jacobi Method; Power Method; Monte Carlo Methods; Conclusions; Benchmarking CGM; Introduction; Additional CGM Description; Heterogeneous Machine; Algorithm Implementation and Timing Results; Conclusions; OpenCL Fundamentals; OpenCL Overview; What is OpenCL; CPU + Accelerators; Massive Parallelism Idea; Work Items and Workgroups; OpenCL Execution Model; OpenCL Memory Structure<br>OpenCL C Language for Programming KernelsQueues, Events and Context; Host Program and Kernel; Data Parallelism in OpenCL; Task |

Parallelism in OpenCL; How to Start Using OpenCL; Header Files; Libraries; Compilation; Platforms and Devices; OpenCL Platform Properties; Devices Provided by Platform; OpenCL Platforms - C++; OpenCL Context to Manage Devices; Different Types of Devices; CPU Device Type; GPU Device Type; Accelerator; Different Device Types - Summary; Context Initialization - by Device Type; Context Initialization - Selecting Particular Device; Getting Information about Context

OpenCL Context to Manage Devices - C++Error Handling; Checking Error Codes; Using Exceptions - Available in C++; Using Custom Error Messages; Command Queues; In-order Command Queue; Out-of-order Command Queue; Command Queue Control; Profiling Basics; Profiling Using Events - C example; Profiling Using Events - C++ example; Work-Items and Work-Groups; Information About Index Space from a Kernel; NDRange Kernel Execution; Task Execution; Using Work Offset; OpenCL Memory; Different Memory Regions - the Kernel Perspective; Relaxed Memory Consistency

Global and Constant Memory Allocation - Host CodeMemory Transfers - the Host Code; Programming and Calling Kernel; Loading and Compilation of an OpenCL Program; Kernel Invocation and Arguments; Kernel Declaration; Supported Scalar Data Types; Vector Data Types and Common Functions; Synchronization Functions; Counting Parallel Sum; Parallel Sum - Kernel; Parallel Sum - Host Program; Structure of the OpenCL Host Program; Initialization; Preparation of OpenCL Programs; Using Binary OpenCL Programs; Computation; Release of Resources; Structure of OpenCL host Programs in C++; Initialization Preparation of OpenCL Programs

Sommario/riassunto

In 2011 many computer users were exploring the opportunities and the benefits of the massive parallelism offered by heterogeneous computing. In 2000 the Khronos Group, a not-for-profit industry consortium, was founded to create standard open APIs for parallel computing, graphics and dynamic media. Among them has been OpenCL, an open system for programming heterogeneous computers with components made by multiple manufacturers. This publication explains how heterogeneous computers work and how to program them using OpenCL. It also describes how to combine OpenCL with OpenGL for displaying graphical effects in real time. Chapter 1 describes briefly two older de facto standard and highly successful parallel programming systems: MPI and OpenMP. Collectively, the MPI, OpenMP, and OpenCL systems cover programming of all major parallel architectures: clusters, shared-memory computers, and the newest heterogeneous computers. Chapter 2, the technical core of the book, deals with OpenCL fundamentals: programming, hardware, and the interaction between them. Chapter 3 adds important information about such advanced issues as double-versus-single arithmetic precision, efficiency, memory use, and debugging. Chapters 2 and 3 contain several examples of code and one case study on genetic algorithms. These examples are related to linear algebra operations, which are very common in scientific, industrial, and business applications. Most of the book's examples can be found on the enclosed CD, which also contains basic projects for Visual Studio, MinGW, and GCC. This supplementary material will assist the reader in getting a quick start on OpenCL projects.